Network Working Group
Request For Comments 578
NIC 19501

Abhay Bhushan, MIT-PTD (DMS)
Neal D. Ryan, MIT-PTD (DMS)
October 8, 1973

USING MIT-MATHLAB MACSYMA FROM MIT-DMS MUDDLE
An Experiment in Automated Resource Sharing

I. INTRODUCTION

This paper describes an experiment in non-trivial automated resource sharing between dissimilar systems. The goal of the experiment was to interface the MUDDLE system at MIT-DMS (Host 70.) to the MACSYMA system at MIT-Mathlab (Host 198.), in such a manner that the MUDDLE-user at MIT-DMS is not required to know anything about the ARPANET, Mathlab, or even MACSYMA. In fact, the user need not be aware that part of the computation is performed by MACSYMA on the Mathlab computer.

This experiment differs from the MATHLAB-UCSB/OLS experiment (ref. NWG/RFC 525, NIC 17161 "MIT-MATHLAB Meets UCSB-OLS" by Parrish and Pickens) in several important respects. First, the use of the remote network resource is completely automated. The human user does nothing more than use a function in MUDDLE such as "INTEGRATE" which requires the remote MACSYMA resource for computation. The program performs all the required tasks of connecting to Mathlab, log in, and using MACSYMA. (In the UCSB-OLS experiment, the user had to manually connect to Mathlab, login, use MACSYMA, type the input in a form suitable for MACSYMA, save the results in a file at Mathlab, disconnect from Mathlab, start a retrieval job at UCSB to retrieve the "saved"

results, and finally submit the results to a local program). Second, the use of the remote resource is completely integrated into the local MUDDLE system. The user can specify the computations in a form that MUDDLE understands. The resource-sharing program (whose existence the user need not be aware of) does the translation from the MUDDLE "prefix" form to the MACSYMA "infix" form on input, and vice-versa on output. This ability allows the MACSYMA resources to be completely integrated into MUDDLE to the extent that parts of the same computation can be performed by MACSYMA and others by MUDDLE.

II. THE MACSYMA AND MUDDLE RESOURCES

Before proceeding to describe the resource sharing facility a description of the two resources, MACSYMA and MUDDLE, is in order. The MACSYMA system at Mathlab is a powerful resource for symbolic manipulation of algebraic functions. It can, among other things, perform symbolic integration and differentiation, expand series, perform Laplace and inverse-Laplace transforms, solve equations and systems of equations, and simplify rational functions. (A description of MACSYMA's capabilities is given in "The MACSYMA Users' Manual" available from the MIT-Mathlab group at Project MAC).

The MUDDLE system provides a general-purpose environment suitable for automatic programming, graphics, data management, "networking", and mathematical computations. The MUDDLE language

represents a powerful extension of the list processing language LISP in the area of data types such as strings, vectors, uniform vectors, and user definable types. (MUDDLE is described in some detail in "The MUDDLE Primer" (SYS.11.01) by Greg Pfister, available from the Programming Technology Division at Project MAC).

Muddle has extensive graphical and numerical computation facilities. The user can display graphs on ARDS and IMLAC type consoles, and on the Evans and Sutherland (E&S) display system. The MUDDLE console graphics provide a facility to view graphical representation of functions with overlay capability and automatic scaling that can be controlled by the user. The E&S provides the user with a versatile tool for studying the dynamic characteristics of graphs, curved surfaces, and other three-dimensional objects. The combination of MACSYMA, MUDDLE, and the E&S graphics capabilities represents a very powerful resource for problem solving that is integrated and made easily usable by the resource sharing facility.

III. THE AUTOMATED RESOURCE-SHARING FACILITY

The resource-sharing facility described herein uses the most easily accessible communication path to MACSYMA, the TELNET connection to the logger service on socket 1. No modifications were made to MACSYMA, nor were any special programs created on the Mathlab computer. The entire task of resource sharing is

performed by programs in MUDDLE. Let us say on the outset that we are not advocating this mode of usage for automated resource sharing. A resource-sharing protocol that allows convenient use of remote resources via programs is far more reliable and efficient way, but that requires work on part of server sites. The existing protocols and systems FTP, RJE, RSEXEC, and the Datacomputer cater to a limited subset of easily managed resources. We register here our desire for uniformity (which alas is lacking) in the current systems, and work along the direction of general-purpose resource sharing. In the absence of a general resource-sharing protocol and a MACSYMA server to go along with it at Mathlab, the TELNET connection is the best a user can do.

The resource sharing facility comprises of several independent but integrated parts. These are:

1) Connecting to Mathlab, login, and invoking MACSYMA.
2) Conversion of Muddle's prefix to MACSYMA's infix form.
3) Generation of MACSYMA input.
4) Interpreting MACSYMA's results including errors and comments.
5) MACSYMA infix to MUDDLE prefix conversion.
6) Plotting graphs for the functions
7) Allowing human intervention if desired.
8) Disconnect from MACSYMA.

The user (assuming that he has loaded the necessary programs in MUDDLE) to integrate the function "3*X" has to only type:

<INTEGRATE '<* .X 3>>$

where '$' represents the ASCII character <ESC> (or <ALT-MODE>). MUDDLE will then return the following result:

</ <* 3 <^ .X 2>> 2>

Alternately, if the user wishes to use infix form, he can type:

<INTEGRATE "3*X">$

and the corresponding answer returned by MUDDLE would be

"3*X^2/2"

The following sequence of events takes place when integrate (or any other function that uses MACSYMA) is used. If the user isn't already communicating with a MACSYMA (the program keeps track of the connection), a connection is established to MIT-Mathlab, the user is logged in (automatically by program, using the user's identification), and a MACSYMA is initiated. A prefix to infix conversion is performed and the following input is sent to MACSYMA (using the above example):

STRING (INTEGRATE (3*X,X));

The program then interprets MACSYMA's output recognizing error responses and comments and extracts the result if no error is encountered. The result which is in infix form is then converted to the prefix form which is returned by the MUDDLE function INTEGRATE.

The INTEGRATE function takes an optional argument, the variable with respect to which the integration is to be performed. The syntax for the function is:

<INTEGRATE <EXPR> ["<VAR>"]>

where EXPR is any expession of the type STRING or QUOTED FORM. The optional argument (in square brackets) VAR must be of the type STRING (enclosed by double-quotes). The syntax of other functions is:

<SIMPLIFY <EXPR>>

<DIFF <EXPR> ["<VAR>" "<TIMES>"]>

<EXPAND <EXPR> ["<MAXPOSEX>" "<MAXNEGEX>"]>

where TIMES is the number of times the EXPR is to be differentiated and MAXPOSEX and MAXNEGEX control the maximum positive and negative integer exponent to be used in expansion. The default value for VAR is "X", for times is "1", and for MAXPOSEX and MAXNEGEX is "6" each.

The user can use the result returned by MUDDLE in any of his computations, including drawing a graph. For example, typing:

<GRAPH <DIFF '<^ .X 3>> X -5 5>$

to MUDDLE will draw the graph "Y = 3*X^2" on the IMLAC or ARDS screen with values of X from -5 to +5 (assuming the user has the graphics package and the right IMLAC program loaded). The same graph would be drawn if the user typed:

<GRAPH <IPARSE <DIFF "X^3">> X -5 5>$

where IPARSE is the MUDDLE function that converts infix to prefix

form. The corresponding function for prefix to infix conversion is UNIPARSE.

The details of using the MACSYMA resource sharing facility may be gathered from the annotated script of the example given in Section V of this paper.

IV. CAPABILITIES AND LIMITATIONS

The program tries to be helpful to the user as much as possible. For example, if for some reason the MIT-Mathlab computer is not available, the MACSYMA service at the MIT-AI computer is procured.

It should be mentioned that though the program is fairly capable in retrieving results, recognizing error messages, and separating comments, its recognition is not fool-proof. The program only makes an educated guess as to where the answer lies, it is not as clever as a human user sitting at a console, who can filter out such messages as "System going down" and communication from another user (console-link) if they were to appear in the middle of the result. This points to one of the pitfalls of using a facility via a program that is basically designed for use by human users.

The program reliability can be marginally improved by asking MACSYMA to print special characters before and after the results it sends (but again this is not fool-proof). For example, the following input to MACSYMA:

```
Block ([ans],
print (/(),
ans: diff (XΛ2,X),
print (string (ans))
print (/)),
return (ans));
```

will cause MACSYMA to generate the following output:

```
(
2*X
)
(D**)          2X
```

From the above output, the answer "2*X" can be easily extracted.

The resource sharing program does however recognize the so-called "unintegratable" functions such as EXP (XΛ2)" - and gives the correct error response. Normally, the user is in "TERSE" mode, and does not see the interaction between MACSYMA and MUDDLE. To see the interaction the user must enter "VERBOSE" mode by typing:

<VERBOSE>$

to MUDDLE. To return to "TERSE" mode the user types:

<TERSE>$

The user can also, if he is profficient in use of MACSYMA, communicate directly with MACSYMA at any point by typing:

<TELCOM 1>$

to MUDDLE. The TELCOM feature may be useful if the user wishes to see what is going on, or wants to examine the MACSYMA computations by entering the LISP environment (typing <Control-G> to MACSYMA). To return to MUDDLE and the automated environment, the user first escapes to MUDDLE by typing <Control-E>, and then

types:

<AUTO>$

to MUDDLE. If the user types "<ERRET 1>$" after escaping to MUDDLE from "TELCOM" mode, he will be returned in direct communication with MACSYMA. If the user discovers that his "MACSYMA" is hopelessly confused or if he wishes to start a new version of MACSYMA, he must type:

<DIS>$

to MUDDLE, which will disconnect him. Typing "<MACSYMA>$" or using any of the functions that use MACSYMA will connect him to MACSYMA again.

Currently, MUDDLE recognizes and takes action as described above whenever differentiate, integrate, expand, simplify, and integrate.simplify (integrate and simplify) functions are encountered. But it is quite easy to generate programs for other operations such as Laplace transforms and solving equations. The prefix-to-infix conversion and vice-versa works for all mathematical forms we have encountered so far in our short experiment.

An alternate way to utilize MACSYMA's capabilities would have been to use it in the LISP environment by constructing a suitable interface between LISP and MUDDLE. Such an approach would avoid the multiple conversions from prefix to infix form and vice-versa, but other, perhaps more difficult, conversions would be required.

V. EXAMPLE

The following scenario describes the use of the resource-sharing facility. The facility is accessible in the MUDDLE system at MIT-DMS. The interaction between MUDDLE and MACSYMA, normally not visible to the user, is also shown here (in VERBOSE mode) so that the reader may gain a better understanding of how the program operates. It should be noted that the graphs will be plotted only if the user has loaded the "graphics package" and is on an IMLAC or ARDS console. We would also like to stress that this scenario is not intended to demonstrate the full capabilities of MACSYMA, or of MUDDLE, but only to illustrate the resource sharing facility.

SCENARIO FOR USING THE MUDDLE-MACSYMA FACILITY

(In the following scenario, user input is underlined and our comments are preceded with a semicolon. ↵ represents a carriage return and $ represents <ESC> or alt-mode. The user is assumed to be logged in at MIT-DMS (Host 70). Note that the input should be typed exactly as shown, as MUDDLE distinguishes between upper and lower case characters. Please refer to "THE MUDDLE PRIMER" (SYS.11.01) by Greg Pfister for a description of the MUDDLE system and to "MUDDLE CONSOLE GRAPHICS USER GUIDE" (SYS.11.11) by

Neal Ryan for description of the graphics package. Both documents are available from the Programming Technology Division at Project MAC.)

```
[;]MUDDLE>                                  ; Get a MUDDLE, ';' is MONIT prompt.
MUDDLE 42 IN OPERATION.
LISTENING-AT-LEVEL 1 PROCESS 1
<FLOAD "MUDDLE;MACSYM">$                    ; Load the program from MUDDLE diretory.
/METMUDGIN GOUT
GIN GOUT                                    ; Harmless comments from MUDDLE.
"DONE"
<DIFF '<- <* .X <LOG .X>> .X>>$
PLEASE BE PATIENT, MACSYMA LOADING MAY TAKE TIME
MACSYMA AT MIT-MATHLAB                      ; Comments from the program.
<LOG .X>                                    ; The result is a MUDDLE form.
<INTEGRATE '<LOG .X>>$
SIN FASL DSK MACSYM BEING LOADED
LOADING DONE                                ; Comments from MACSYMA
SCHATC FASL DSK MACSYM BEING LOADED
LOADING DONE
<- <* .X <LOG .X>> .X>                      ; The answer again.
<SET A <INTEGRATE "X/(XΛ3+1)">>$            ; The input is in infix form.
"LOG(XΛ2-X+1)/6+ATAN((2*X-1)/SQRT(3))/SQRT(3)-LOG(X+1)/3"
                                            ; The answer now is in infix form.
<SET B <DIFF .A>>$
"2/(3*((2*X-1)Λ2/3+1))+(2*X-1)/(6*(XΛ2-X+1))-1/(3*(X+1))"
<SIMPLIFY .B>$
"X/(XΛ3+1)"                                 ; We get back the original expression.
<EXPAND '<Λ <+ .X 2> 5>>$
<+ <+ <+ <+ <+ <Λ .X 5> <* 10 <Λ .X 4>>> <* 40 <Λ .X 3>>>
<* 80 <Λ .X 2>>> <* 80 .X>> 32>
<INTEGRATE '<EXP <Λ .X 2>>>$
RISCH FASL DSK MACSYM BEING LOADED
LOADING DONE
*ERROR*                                     ; Program recognizes that MACSYMA
CANT-INTEGRATE                              ; couldn't integrate.
LISTENING-AT-LEVEL 2 PROCESS 1
<ERRET>$                                    ; To get back to level 1.
LISTENING-AT-LEVEL 1 PROCESS 1
<DIS>$              ; We disconnect here to show the verbose mode,
                    ; the program disconnects automatically on quitting.
"CONNECTIONS CLOSED NOW"
<VERBOSE>$
"YOU WILL BE ABLE TO OBSERVE MUDDLE-MACSYMA INTERACTION NOW"
<DIFF '<Λ .X 3>>$
```

```
PLEASE BE PATIENT, MACSYMA LOADING MAY TAKE TIME
MIT MATHLAB PDP-10 STELNT.59
ML ITS.1. DDT.516.
10. USERS
:LOGIN 70GUEST              ; The program uses User's SNAME (GUEST here).
:MACSYMA
THIS IS MACSYMA 226
SEE UPDATE > MACSYM; FOR CHANGES
FIX 226 DSK MACSYM BEING LOADED
LOADING DONE
(C1)
MACSYMA AT MIT-MATHLAB              ; The program announces MACSYMA,
 STRING (DIFF ((X^3),X,1));         ; and sends input in infix form.
(D1)                                        3*X^2
<* 3 <^ .X 2>>                       ; The output is in MUDDLE prefix form.
<INTEGRATE '</ .X <+ .X 1>>>$
C2) STRING (INTEGRATE ((X/(X+1)),X));
SIN FASL DSK MACSYM BEING LOADED
LOADING DONE
SCHATC FASL DSK MACSYM BEING LOADED
LOADING DONE
(D2)                                        X-LOG(X+1)
<- .X <LOG <+ .X 1>>>               ; The output again.
<TERSE>$
"OK"                                ; Back in TERSE mode now.

<FLOAD "MUDDLE;UGRF">$              ; To load graphics program
IMLAC? (ANSWER Y OR N) Y            ; for graphics on an IMLAC.
"DONE"
<GRAPH <SLT A '<^ <SIN .X> 2>> X -3 3>$
        ; To graph function sin(X)^2 (graph 1 on Figure 1).
<GRAPH  <DIFF .A>>$
        ; To graph diff of sin(X)^2 (see graph 2, Figure 1).
<GRAPH <INTEGRATE .A>>$
        ; To graph integral of sin(X)^2 (see graph 3, Figure 1).
<QUIT>$                             ; To quit from program and MUDDLE.
KILL
[;]                         ; semicolon prompt from MONIT.
```
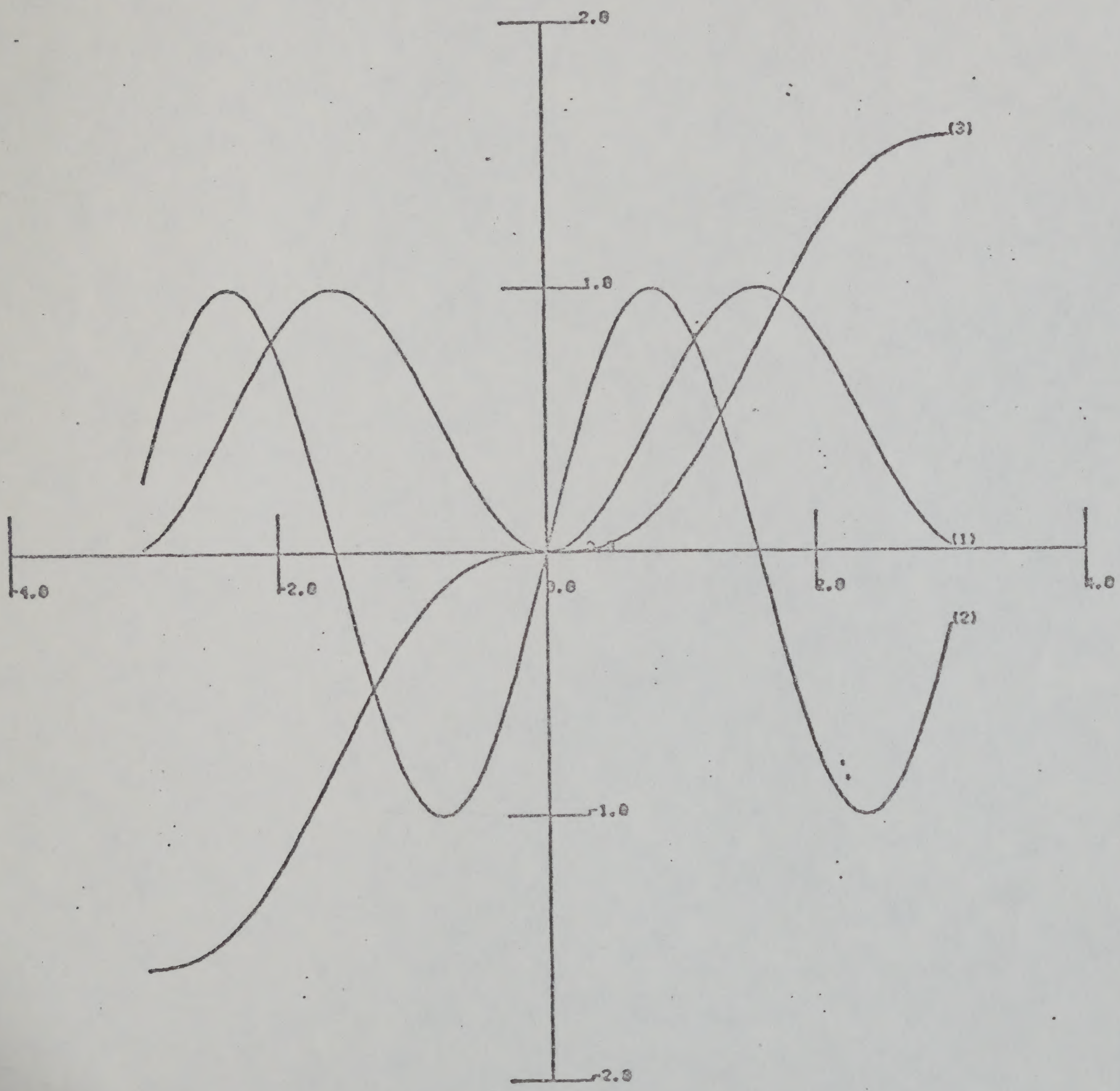
FIG 1. GRAPH FOR SIN(X)^2, DIFF(SIN(X)^2), AND INTEGRATE(SIN(X)^2)